**УДК 51.001.57+004.652.4+004.827**

**М.О. МЕДИКОВСЬКИЙ, Н.Б. ШАХОВСЬКА**

# ФОРМАЛІЗАЦІЯ ОПЕРАЦІЙ НАД ДЖЕРЕЛАМИ ДАНИХ У ПРОСТОРІ ДАНИХ

***Національний університет «Львівська політехніка»,***
***Львів, Ст. Бандери, 12,***
***E-mail: natalya233@gmail.com***

**Анотація.** Побудовано формальну модель простору даних та уведено операції над ним. Розроблено операції роботи з різнотипними даними.

**Аннотация.** Проанализированы проблемы обработки данных с различных источников. Описана формальная модель пространства данных и операции над ним.

**Abstract.** Problems which arise up during work with separate sources with depositories information using and databases are analyzed. There are formalized model of dataspace as mean of integration and information analysis from separate sources.

**Ключові слова:** простір даних, алгебраїчна система, сховище даних, база даних, інтеграція.

## ВСТУП

Інформаційне суспільство – суспільство, у якому створення, передавання, дифузія, використання, інтеграція і маніпулювання інформації – важлива господарська, політична і культурна діяльність [Вікіпедія]. Специфікою до цього виду суспільства є те, що інформаційна технологія є центральною позицією для виробництва, економіки, суспільства загалом. Інформаційне суспільство вважають наступником індустріального суспільства.

У сучасному суспільстві інформація стає найдорожчою цінністю, а індустрія отримання, опрацювання і трансляції інформації – провідною галуззю діяльності, куди з кожним роком вкладають все більші капітали. Як вважають провідні вчені, інформація стає важливим стратегічним ресурсом, відсутність якого призводить до суттєвих втрат в економіці. Інформатизація суспільства виступає одним з вирішальних чинників модернізації економіки на ринкових засадах і запорукою інтеграції України у світове співтовариство.

Для прийняття адекватних рішень у певній галузі необхідно, щоб дані, які надходять із різних джерел і використовуються для прийняття керівних рішень, задовольняли такі вимоги:

- були повними, несуперечливими та надходили вчасно;
- були інформативними, оскільки вони застосовуватимуться для прийняття рішень;
- були однакової структури, щоб мати можливість завантажити їх у єдине сховище даних та проаналізувати;
- зберігалися в однакових моделях даних та були незалежними від платформи розроблення, щоб мати можливість використання цих даних в інших засобах.

Однак, на сьогодні немає жодної методики опрацювання даних, яка б задовольняла всі наведені вимоги до опрацювання даних, а отже, немає можливості аналізувати стан галузі загалом, використовуючи першоджерела інформації, а не визначені наперед статистичні звіти. Так, наприклад, для керівництва туристичною галуззю використовуються результати аналізу зведеної форми 1 Тур та надходжень з митниць. Така наявна інформація дозволяє фіксувати факт настання певної причини та її наслідки, але найчастіше не дозволяє визначати причини, оскільки для аналізу використовується обмежена і наперед жорстко визначена частина інформації.

За останні роки спостерігається зростання потреби в «даних, які застосовуються у всіх сферах», що привело до виникнення нового типу інформаційної інтелектуальної системи. Сьогодні найгостріші проблеми керування інформацією виникають в організацій (наприклад, готелів, баз відпочинку, оздоровчих закладів, туристичних агентств), робота яких полягає в опрацюванні великої кількості різнотипних, взаємонезалежних джерел даних. Такий тип системи отримав назву *простір даних*. На

відміну від систем інтеграції даних, що також пропонують загальноприйнятий доступ до різнорідних джерел даних, простори даних не припускають, що всі семантичні взаємозв'язки між джерелами відомі і вказані. Багато користувачів, які працюють з просторами даних проводять дослідження даних, і немає єдиної схеми, по якій вони можуть створювати запити. Тому, важливо, що запити є дозволеними елементами, щоб конкретизувати різні ступені структури, а використання ключового слова робить запит більш структурованим.Стаття присвячена поданню простору даних як алґебраїчної системи та формалізації операцій над ним.

## ПОСТАНОВКА ПРОБЛЕМИ

У різних галузях науки спостерігається експоненційний ріст обсягів експериментальних даних. Складність використання таких даних виникає внаслідок їхньої природної різнорідності (зберігання у різних системах, призначення для різних задач, різні методи опрацювання та зберігання тощо).

Розрив, який збільшується між джерелами даних і сервісами, приводить до необхідності пошуку нових шляхів організації рішення задач над множинними розподіленими колекціями даних і програм, які концентруються в спеціалізованих центрах даних і обчислювальних ресурсах.

Традиційно при рішенні певних задач фахівці використовують звичні для них джерела інформації і формулюють завдання з огляду на лише такі джерела. Очевидна неповнота інформації, яку вдається охопити при такому підході. Безліч джерел даних і сервісів, що існують в Інтернеті, їхня розмаїтість, викликають потребу в радикальній зміні такого традиційного підходу. Сутність цієї зміни полягає в тому що задачі повинні формулюватися незалежно від існуючих джерел інформації, і лише після такого формулювання повинна здійснюватися ідентифікація релевантних завданню джерел, приведення їх до виду, необхідному для розв’язання задачі, інтеграція, ідентифікація сервісів, які дозволяють реалізувати окремі частини абстрактного процесу рішення завдання.

**Метою статті** є розроблення методології сумісного використання та аналізу вмісту різнорідних джерел інформації для прийняття керівних рішень.

**Наукова новизна** полягає у формалізації опису простору даних як алгебраїчної системи та уведенню множин операцій та відношень над носіями даних алгебраїчної системи типу «простір даних», що дозволило розробити схеми мета даних для опису джерел та побудувати алгоритми роботи з ними.

**Практична цінність** полягає у розробленні агента визначення структури джерела даних, який дозволяє модифікувати запити користувачів та налаштовувати їх стосовно типу конкретного джерела.

**Постановка задачі.** У статті опишемо основні елементи простору даних, формалізуємо методи взаємодії між ними та розробимо методи автоматичного визначення структур даних джерела. Доведемо, що сховища даних та бази даних є алгебраїчними системами та елементами простору даних одночасно.

## АНАЛІЗ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

На сьогодні немає жодної методики опрацювання даних, яка б задовольняла всі наведені вимоги до опрацювання даних, а отже, немає можливості аналізувати стан галузі загалом, використовуючи першоджерела інформації, а не визначені наперед статистичні звіти [1, 2]. Розроблені методи інтеграції даних спираються на джерела даних із наперед визначеними структурами, які мають відомі механізми погодження [3, 4], що є неприпустимим у разі прийняття керівного рішення по усій предметній області.

Простір даних розглядають як нову абстракцію керування даними [4]. Основоположником ідеї просторів даних був Алон Хелеві. Ведуться два проекти, орієнтовані на підтримку просторів індивідуальних даних. Перший з них – проект SEMEX (SEMantic Explorer) [5,6], виконується в University of Washington під керівництвом Хелеві. Другий, з назвою iMeMex [7], виконується під керівництвом Йенса-Петера Диттриха в ETH Zurich. Проте, судячи з аналізу Інтернет-джерел, жоден з проектів ще не формалізував поняття простору даних, що, у свою чергу, призводить до розрізненості підходів роботи з ними.

Важливим елементом інтеграції є сумісне використання структурованих, частково структурованих та неструктурованих джерел інформації. Як показано у [7], наразі проблема пошуку напівструктурованої інформації вирішується лише в окремих областях, для яких побудована онтологія.

## ОСНОВНИЙ МАТЕРІАЛ

Дослідження в області моделей даних інформаційних систем [3,4] показують, що на сьогодні центральним стало поняття типу даних. З цим зв'язані як проблематика створення нових мов програмування, так і впровадження сучасних технологій організації даних, зокрема, і просторів даних.

Будь-який інформаційний простір $E$ доцільно подати у вигляді абстрактної алгебраїчної системи

***A= <Al, WF, WR>*** [8], де ***Al*** – об'єкти інформаційного простору; ***WR*** – зв'язки між об'єктами ***Al***; ***WF*** – операції маніпулювання об'єктами у просторі. Як об'єкти моделі (1) можуть виступати компоненти інтелектуальної системи – файли всіх типів, каталоги, логічні і фізичні диски.

Відношення ***WR*** =$\{ WR1, \ldots WR_n1 \}$ між об'єктами інформаційного простору визначає конкретну конфігурацію інтелектуальної системи, орієнтовану на конкретного користувача чи користувачів, ***G***=$\{G1 , \ldots G_n2 \}$ – множина користувачів. Модель взаємодії користувача з інформаційним простором можна подати у вигляді:

$$Y(t)=E\,(Z1(t), \ldots, Z_n2(t)),$$

де $Z_i(t)$ – вхідний вплив на інформаційний простір з боку користувача $G_i \in \boldsymbol{G}$; $Y(t)$ – реакція системи (відповідь), що сконфігурована під користувача і має вигляд $E$. У загальному випадку $Z_i(t)$ – елементарна задача, що користувач $G_n$ вирішує за допомогою інформаційного простору $E$(***Al***, ***WR***, ***WF***). Прикладами елементарних задач є: пошук інформації (за зразком, за індексом, за описом, за методом найближчого сусіда тощо), інтеграція даних (консолідація, федералізація, розповсюдження), агрегація тощо [9].

У загальному випадку кожна із елементарних задач вирішується на певному носії даних $Al_j$, $j$=1, …, $n$3, із використанням певних операції маніпулювання $WF_j$, ефективність виконання яких для задачі $Z_i(t)$ залежить від типу носія. Користувач не знає наперед, з яким саме носієм йому потрібно працювати, та дозволені операції над цим носієм. Тому визначення типу елементарної задачі відбувається за допомогою множини відношень ***WR***.

Множина відношень ***WR*** здійснює структуризацію знань про носій інформаційного простору та допустимі операції над ним.

Визначимо правила структуризації даних довільної предметної області:

- факторизація множини об'єктів інформаційного простору ***Al*** за відношенням еквівалентності [3];
- конструювання додаткових функцій *Id*, *Num, Selector*;

*Id*($x$) – функція задає для кожного об'єкту додатковий атрибут – його індивідуальний ідентифікатор,

*Num*($x$) –функція задає для кожного об'єкту додатковий атрибут – його порядковий номер в класі еквівалентності $X_i$, де $i$ = 1 ., $p$. Областю значення функції *Num* є множина натуральних чисел,

*Selector*($x$) – функція задає для кожного об'єкту додатковий атрибут – його подання. Областю значень для цієї функції є деякий кортеж з атрибутів об'єкту, тобто значень функцій *Id*($x$), *Num*($x$), $f_1(x), f_2(x), ..., f_k(x)$

- побудова інвертованих індексів [4];
- побудова багатовимірних матриць (використання алґебри кортежів).

Оскільки інструмент моделювання баз даних повинен з потреби включати не лише засоби структуризації даних, але і операційні можливості для маніпулювання даними, модель даних в інструментальному сенсі розуміється як алґебраїчна система.

Основними моделями для побудови інформаційних систем є бази даних, сховища даних, простори даних. Подамо кожен із зазначених об'єктів як алгебраїчну систему.

Отже, реляційна база даних – це алгебраїчна система, у якої носієм є множина реляційних відношень ***r***, множиною операцій – реляційна алгебра $\Re$, множиною предикатів – словник даних (схема даних бази даних) ***R***.

$$\boldsymbol{DB}=<\boldsymbol{r},\ \Re\ ,\boldsymbol{R}>, \qquad (1)$$

$$\Re = \{\pi, \sigma, \triangleright\triangleleft \cup, \cap, -\}.$$

Тепер дамо формальне означення сховища даних.

*Сховищем даних* (СД) назвемо шістку

$$DW=<DB,rf,RF,rm,RM,func>,$$

де **DB** – множина вхідних баз даних (реляційних, багатовимірних, об'єктно-орієнтованих,

ненормалізованих тощо) (або множина відношень, їх схем та обмежень цілісності, які містять інформацію з вхідних баз даних), ***rf*** – множина відношень фактів, ***RF*** – схема ***rf***, ***rm*** – множина відношень метаданих, ***RM*** – схема ***rm***, ***func*** – множина процедур прийняття рішень.

Тоді *нові дані* (або *рішення*) – це результат застосування функцій сховища даних над відношенням фактів:

$$Design = func(rf, user\_param),$$

де *user_param* – множина параметрів користувача, або вимог, які ставляться до рішення.

*Відношення між вимірами* – відношення, яке є зв'язком між певними вимірами та відношенням фактів: $V_1 \times V_2 \times ... \times V_n \times rf \to rel$ .

У відношенні фактів виміри подаються за допомогою зовнішніх ключів, а самі значення – за допомогою атрибутів агрегації. У свою чергу, *rel* можуть бути параметрами для інших відношень між вимірами і тим самим створювати ієрархію вимірів.

Над даними сховища даних виконуються такі операції:

1. *Інтеграція даних* – це об'єднання даних, які знаходяться у різних системах (базах даних). Існують такі методи інтеграції:

консолідація даних – це збір даних з територіально віддалених або різноплатформенних джерел $DB_i$ даних в єдине сховище даних DW з метою їх подальшого опрацювання та аналізу.

$$\mathrm{DW}.rel \xrightarrow{consolid} \mathrm{DB}_1.r \cup \ldots \cup \mathrm{DB}_n.r.$$

операція *федералізації даних* полягає у витяганні даних з первинних систем на підставі зовнішніх вимог. Всі необхідні перетворення даних здійснюються при їх витяганні з первинних файлів.

$$\text{Virtual.DW}: \sigma_{fed\ \mathrm{rm=DB1.r}}(\mathrm{DB}_1.\mathrm{r}) \cup \ldots \cup \sigma_{fed\ \mathrm{rm=DB\,n.r}}(\mathrm{DB}_n.\mathrm{r}).$$

2. *Агрегація даних* – це обчислення узагальнених значень на основі даних відношень вимірів для підтримки стратегічного або тактичного керування з детальних даних.

$$\boldsymbol{rel}=Ag(\mathrm{DB1}.\boldsymbol{r},\ldots,\mathrm{DBn}.\boldsymbol{r}).$$

Опишемо сховище даних як алгебраїчну систему.

Оскільки воно інтегрує інформацію з баз даних, а інтегровані значення містяться у відношенні фактів, то звідси випливає, що сховище даних – це алгебраїчна система виду

$$\mathbf{DW}=< \boldsymbol{DW}, \aleph, \boldsymbol{rm} >, \tag{2}$$

$$\boldsymbol{DW}=\{\boldsymbol{rel}, \mathrm{DB}_1.\boldsymbol{r},\ldots,\mathrm{DB}_n.\boldsymbol{r}\},$$

$$\aleph = \{\Re, \xrightarrow{consolid}, \sigma_{fed}, Ag, func\}.$$

Отже, алгебраїчна система класу реляційна БД є підсистемою алгебраїчної системи класу сховище даних.

Тепер дамо формальне означення простору даних.

*Простір даних* **DS** – це множина даних, поданих у різних моделях (баз даних **DB**, сховищ даних **DW**, статичних Web-сторінок **Wb**, неструктурованих даних **Nd**, графічних та мультимедійних даних **Gr**), локальних сховищ та **ODW**, а також засобів інтеграції **Int**, пошуку **Se** та опрацювання інформації **Wo**, об'єднаних середовищем керування моделями **EM** [10,11].

$$\mathrm{DS}=<\mathrm{DB}, \mathrm{DW}, \mathrm{ODW}, \mathrm{Wb}, \mathrm{Nd}, \mathrm{Gr}, \mathrm{Int}, \mathrm{Se}, \mathrm{Wo}, \mathrm{EM}>. \tag{3}$$

Каталог **CG** – це реєстр ресурсів даних, що містить найбільш базову інформацію про кожного з них: джерело, ім'я, місцезнаходження в джерелі, розмір, дату створення і власника та ін. Каталог є інфраструктурою для більшості інших сервісів простору даних, але він також може підтримувати базовий, призначений для користувача, інтерфейс проглядання простору даних.

Для ораганізації робот из розрізненими дежерелами використовуютьс словник термінів та понять (ключових слів) *Dic*, який містить синонімічний опис одного і того ж концепту у різних джерелах даних. Заповнення словника даних на початку здійснюється за допомогою розробленої онтології предметної області, пізніше – автоматизовано.

$$\text{Metadata}(\mathbf{DB}, \mathbf{DW}, \mathbf{Wb}, \mathbf{Nd}, \mathbf{Gr, ODW}) \cup Dic \Rightarrow \mathbf{Cg}. \quad (4)$$

Для подання простору даних як алгебраїчної системи необхідною умовою є уніфікація джерел даних, оскільки саме вони є носіями (об'єктами, наж якими виконуються операції та відношення алгебраїчної системи). Уніфікація сховищ даних та баз даних здійснюється за допомогою інтелектуального агента (подано нижче). Проте, як видно із визначення простору даних (3), джерелами його інформації є також неструктурований текст та веб-сайти. Для ефективного пошуку та аналізу неструктурованої текстової інформації використаємо семантичну мережу.

Семантична мережа – це структура для подання знань у вигляді вузлів, з'єднаних дугами. Особливості структури семантичних мереж: вузли семантичних мереж являють собою концепти предметів, подій, станів, які у свою чергу визначаються із словника *Dic*; довільні вузли одного концепту відносяться до різних значень, якщо вони не відмічені, що вони відносяться до одного концепту; дуги семантичних мереж створюють відношення між вузлами-концептами (помітки над дугами вказуватимуть на тип відношення).

Семантична мережа, побудована на основі аналізу термів напівструктурованого джерела інформації $Q$, подається як двійка

$$Q = \{V, D\},$$

де $V = \{v_i\}$ - множина вершин (вузлів мережі), $V \in Dic$, $D = \{d_j\}$ - множина дуг.

Дуги між елементами визначають взаємозв'язки між вершинами і задають послідовність пошуку концептів (їх важливість). Вершини є елементами локального сховища даних **ODW.**

Для опису Веб-ресурсів використовують поняття семантичної павутини, функції та структура якої спів мірні з семантичною мережею. Для створення зрозумілого комп'ютеру опису ресурсу в семантичній павутині використовується формат RDF. Оскільки джерелами даних простору даних є Веб-ресурси, то для *Dic* використовуватимемо формат RDF. Пошук у такій мережі здійснюватиметься за допомогою ключових слів.

Побудуємо функцію трансформації напівструктурованого тексту та Веб-сайтів у вигляді семантичної мережі:

SemNet(**Wb**)→**ODW–** для Веб-ресурсів,

SemNet(**Nd**)→**ODW–** для текстових даних.

Подання напівструктурованих даних у вигляді семантичної мережі із збереженням вершин та відношень між ними у локальному сховищі **ODW** дозволяє звести інформацію з неоднорідних джерел даних до баз даних та сховищ даних, що, у свою чергу, при визначенні та уніфікації їхніх стурктур даних дасть можливість здійснювати інтеґрацію, пошук та аґреґування даних.

Визначення структур даних джерел просторів даних здійснюється за допомогою інтелектуального аґента

$$\text{EM}(\mathbf{CG}) \xrightarrow{Agent} \mathbf{ODW}. \quad (5)$$

Аґент *Op* подається сімкою об'єктів [11]:

$$\text{Agent} = < \mathbf{CG}, \text{EM}, \text{Dic}, \text{Experience\_Base}, \text{Solver}, \text{Effector} >, \quad (6)$$

де **G** – ідентифікатор внутрішнього стану аґента (інформація про джерела, що вже є у ПД);

EM – компонента аґента, що відповідає за сприйняття середовища (сенсор), тобто середовище

керування моделями;

*Dic* – база знань, що містить знання аґента про власні можливості (терміни-синоніми, що позначають у джерелах одні і ті ж властивості);

*Experience_Base* – база накопиченого досвіду аґента, що містить "історію" впливів на аґент з боку середовища й відповідної їм реакції аґента (*Experience_Base*= $\sigma_{evdate=Date()}(Dic)$);

*Solver* – компонента, що відповідає за навчання (подає список розбіжностей, які виявив аґент);

*Effector* – компонента, яка відповідає за дії аґента (формування запиту по декількох джерелах, приведення результатів запитів по джерелах до єдиної структури, відмова у запиті).

В основі роботи аґента лежить інформація про джерела, які вже є у просторі. Його задачею є порівняння структур даних джерела даних, що входитиме у простір, з структурами дажних джерел, що вже є у просторі, та визначення різниці. Це дозволить автоматизувати формування запитів, що виконуватимуться у просторі даних. Чим більше джерел здатний «розрізнити» аґент, тим точніше буде інформація в **ODW** і тим ефективніше можна буде проводити процедури інтеграції, пошуку та опрацювання даних у просторі даних **DS**.

Розглянемо принцип роботи аґента порівняння інформації із двох схем даних для тих самих фізичних сутностей. При цьому допускається, що схеми мають різні системи кодування, тобто той самий об'єкт може мати в цих схемах різні ідентифікатори. Допускається, що назви таблиць, атрибутів і розподіл атрибутів по таблицях можуть розрізнятися. Але передбачається, що між схемами існують взаємозв'язки, які можуть бути задані експертами (словник *Dic*). Необхідно класифікувати типи можливих взаємозв'язків і знайти необхідні умови для інтеграції даних на основі цих взаємозв'язків.

Нехай деяка сутність описується в першій схемі даних відношенням *A*, що містить кортежі $\{x_1, x_2, ..., x_n\}$, а в другій схемі даних відношенням *B*, що містить кортежі $\{y_1, y_2, ..., y_m\}$. Відношення *A* і *B* можуть бути як окремими таблицями в реляційній схемі даних, так і переглядами. Запишемо формально умову, що *A* і *B* містять ті самі фізичні сутності. Будемо вважати, що в цьому випадку існують взаємозв'язки між окремими атрибутами $x_i$ й $y_j$. Розглянемо різні типи таких взаємозв'язків між двома скалярними атрибутами *x* і *y*, визначеними на скінчених доменах *X* і *Y* відповідно.

1. Змістовний взаємозв'язок доменів. Найзагальнішим типом взаємозв'язку можна вважати випадок, коли ми хоча б можемо визначити, чи співпадають об'єкти по атрибутах *x* і *y* або не співпадають і чи співпадають назви-синоніми у словнику термінів *Dic*. Інакше кажучи, задана функція змістовної еквівалентності: $P: X \times Y \to \{0,1\}, Dic_{X=Y}$. $P(x, y) = 1$, якщо по атрибутах *x* і *y* об'єкти співпадають, $P(x, y) = 0$ у іншому випадку. Якщо $P(x, y) = 1$ і $Dic_{X \neq Y}$, то доповнюємо *Dic* новими синонімами.

2. Існує відображення, що конвертує *X* в *Y*, якщо для будь-якого $x \in X$ значення існує $y \in Y$ значення , таке що по атрибутах *x* і *y* об'єкти будуть співпадати. Інакше кажучи, існує відображення $F: X \to Y$ таке, що для всіх $x \in X$ виконується рівність

$$P(x, F(x)) = 1, Dic_{X \neq Y}. \tag{7}$$

3. Існує узагальнююче відображення з *X* в *Y* (*Y* – узагальнення *X*), якщо для будь-якого значення $x \in X$ існує рівно одне значення $y \in Y$, таке що по атрибутах *x* і *y* об'єкти будуть співпадати. Інакше кажучи, існує відображення $F: X \to Y$, таке що для всіх $x \in X$ виконуються умова (7) і нерівність

$$P(x, y) < 1, Dic_X, Dic_Y \text{ для всіх } y \neq F(x). \tag{8}$$

4. Існує узагальнююче відображення *X* на *Y* (*X* – деталізація *Y*), якщо для будь-якого значення $x \in X$ існує рівно одне значення $y \in Y$, і для будь-якого у існує хоча б одне значення *x*, таке що по атрибутах *x* і *y* об'єкти будуть співпадати. Інакше кажучи, існує відображення $F: X \to Y$, таке що для всіх $y \in Y$ існує $x \in X$, такий що $F(x) = y$; і для всіх $x \in X$ виконуються умови (7) і (8).

Будемо вважати, що об'єкт, заданий кортежем $a = \{x_1, x_2, ..., x_n\}$ в одній схемі даних, співпадає з об'єктом, заданим кортежем $b = \{y_1, y_2, ..., y_m\}$ в іншій схемі даних, якщо вони співпадають за всіма взаємозалежними атрибутами, тобто для всіх функцій взаємозв'язку відношень $P_{ij}: X_i \times Y_j \to \{0,1\}$ правильна рівність $P_{ij}(x_i, y_j) = 1$. Множину пар індексів (*i*, *j*), для яких задані функції $P_{ij}$, позначимо

$\Omega = \{(i, j)\}$ , $i=Num(x)$, $j=Num(y)$, $x, y \in Dic$ . Тоді можна задати функцію відповідності об'єктів $P : A \times B \to \{0,1\}$ таким чином:

$$P(a,b) = 1, \text{ якщо } P_{ij}(x_i, y_j) = 1 \text{ для всіх } (i, j) \in \Omega; \tag{9}$$

$$P(a,b) = 0, \text{ якщо існує } (i, j) \in \Omega, \text{ такі що } P_{ij}(x_i, y_j) \neq 1 . \tag{10}$$

Отже, результатом роботи аґента є встановлення взаємозв'язку між схемами даних.

Одною із ключових задач побудови простору даних є визначення виразної потужності запитів із **Se**. Над носіями простору даних виконуються такі операції із множини **Se**:

1) *Запит про довільні дані* $\mathrm{Se}_{\mathrm{simple}}$ - у користувачів повинна бути можливість запиту будь-якого елементу даних, незалежно від його формату і моделі даних. Здійснюється на основі ключових слів key_word та каталогу даних CG.

$$\mathrm{Se}_{\mathrm{simple}} : \sigma_{key\_word(Cg)} . \tag{11}$$

2) *Стуктуровані запити* будуються з використанням SQL та подібних мов. За допомогою каталогу визначається, чи джерело, у якому здійснюватиметься пошук, містить структуровану інформацію. Якщо це так, то виконується запит безпосередньо до джерела даних. У іншому випадку запит продовжується виконуватись по каталогу даних у вигляді пошуку ключових слів.

$$\mathrm{Se}_{\mathrm{structured}} : \sigma_{key\_word(Cg)}, \sigma(Source) . \tag{12}$$

3) *Запити до метаданих* повинні забезпечуватися можливості:

- отримання даних про джерело відповіді та місцезнаходження джерела;
- визначення елементів даних в просторі даних, що можуть залежати від заданого елементу даних, і підтримка гіпотетичних запитів;
- визначення рівня невірогідності відповіді.

$$\mathrm{Se}_{\mathrm{meta}} : \sigma_{user\_param}(Cg) , \tag{13}$$

де user_param – множина параметрів користувача (вимог до запиту), його профілю, або вимог, які ставляться до рішення.

Простір даних є не тільки засобом обміну даним. Він повинен містити засоби отримання нових знань. У контексті просторів даних знання – це результат застосування засобів опрацювання даних над джерелами та каталогом даних :

Design=Wo(DB, DW, Wb, Nd, Cg, user_param).

Під профілем користувача будемо розуміти підмножину каталогу даних, яка вказує на ті джерела даних, до яких користувач має доступ.

$$\mathrm{profile} : \sigma_{\mathrm{access=Yes}}(\mathbf{Cg}) .$$

Із визначення простору даних випливає подання ПД як алгебраїчної системи:

$$\mathrm{DS}=< DS, \wp , \mathrm{Cg}>, \tag{14}$$

$$\boldsymbol{DS}=\{\mathbf{ODW}.\boldsymbol{r}, \mathrm{DW1}.\boldsymbol{rel},\ldots,\mathrm{DWn}.\boldsymbol{rel}, \mathrm{SemNet}(\mathbf{Wb}), \mathrm{SemNet}(\mathbf{Nd})\},$$

$$\wp = \{Agent(\aleph), Se_{simple}, Se_{structured}, Se_{meta}, \sigma_{access}, Agent\} .$$

Таке визначення ґрунтується на таких висновках:

- базу даних можна вважати виродженим сховищем даних (сховище даних з єдиним джерелом та обмеженою множиною операцій – реляційною алґеброю);
- оскільки інформація про інші джерела простору даних (**Wb, Nd, Gr**) міститься у каталозі **Cg** (побудова семантичної мережі), а дані, що отримуються з цих об'єктів, за допомогою операцій інтеґрації потрапляють у локальне сховище даних **ODW**, то в просторі даних **Wb, Nd, Cr** можна замінити каталогом даних **Cg**.

Отже, алґебраїчна система класу сховище даних та алґебраїчна система класу реляційна база даних є підсистемами алгебраїчної системи класу простір даних.

Простори даних можуть вкладатися одне в інше (наприклад, простір даних району вкладається в простір даних області), і вони можуть перекриватися (наприклад, простір даних в сфері туризму перекривається з просторами даних оздоровчо-лікувальної, історичної сфери та сфери управління природніми ресурсами). Тому в просторі даних повинні міститися правила розмежування доступу. Прикладами таких розмежувань для простору даних в сфері туризму є: для учасників простору даних в сфері туризму надати можливість пошуку даних у просторах даних оздоровчо-лікувальної, історичної сфери та сфери управління природніми ресурасми; для учасників простору даних сфери управління природніми ресурсами надати права блокування записів та встановлення властивості неактуальності для даних простору даних в сфері туризму та ін.

Уведемо операцію об'єднання просторів даних:

$$DS_1 \cup DS_2 = <\mathbf{DB}_1 \cup \mathbf{DB}_2, \mathbf{DW}_1 \cup \mathbf{DW}_2, \mathbf{Wb}_1 \cup \mathbf{Wb}_2, \mathbf{Nd}_1 \cup \mathbf{Nd}_2, \mathbf{Mp}_1 \cup \mathbf{Mp}_2, \mathbf{ODW}_1 \cup \mathbf{ODW}_2, \mathbf{Int}, \mathbf{Se}, \mathbf{Wo}_1, \mathbf{Wo}_2, \mathbf{EM}>,$$

$$\mathbf{Cg} = \text{profile}(\mathit{Agent}(\mathbf{Cg}_1) \cup \mathit{Agent}(\mathbf{Cg}_2)),$$

$$\mathbf{Int} = \mathbf{Int}_1 = \mathbf{Int}_2,$$

$$\mathbf{Se} = \mathbf{Se}_1 = \mathbf{Se}_2,$$

$$\mathbf{EM} = \mathbf{EM}_1 = \mathbf{EM}_2.$$

Уведемо операцію перетину просторів даних:

$$\mathbf{DS_1 \cap DS_2 = <DB_1 \cap DB_2, DW_1 \cap DW_2, Wb_1 \cap Wb_2, Nd_1 \cap Nd_2, Mp_1 \cap Mp_2, ODW_1 \cap ODW_2, Int, Se, Wo, EM>}$$

$$\mathbf{Cg = Cg_1 \cap Cg_2,}$$

$$\mathbf{Wo = Wo_1 \cap Wo_2,}$$

$$\mathbf{Int = Int_1 \cap Int_2,}$$

$$\mathbf{Se = Se_1 \cap Se_2,}$$

$$\mathbf{EM = EM_1 = EM_2.}$$

## ВИСНОВКИ

У статті подано формальну модель простору даних на базі алгебраїчної системи Мальцева.. Показано, що алгебраїчні системи класу база даних та сховище даних є підкласом алгебраїчної системи класу простір даних. Показано, що простори даних можуть накладатися один на одного. Уведено операції над просторами даних. **Подальші дослідження** стосуватимуться формалізації методів пошуку неструктурованих, напівструктурованих та строго структурованих даних та побудові відповідних алгоритмів.

## СПИСОК ЛІТЕРАТУРИ


1. Интеграция данных и Хранилища. –2005, Електронне джерело: http://citcity.ru/12101/
2. Интеграция корпоративной информации: новое напрямление. – 2005, Електронне джерело:http://citcity.ru/11155/
3. Indexing Relational Database Content Offline for Efficient Keyword-Based Search //Qi Su, Jennifer Widom //9th International Database Engineering №38; Application Symposium (IDEAS'05), 2005.- P. 297-306
4. А. В. Аграновский Индексация массивов документов //А. В. Аграновский, Р. Э. Арутюнян. - Електронне джерело: http://www.scandocs.ru/page.jsp?pk=node_1185787748359.
5. The Wikipedia XML Corpus //L. Denoyer and P. Gallinari. - SIGIR Forum, 2006.
6. DBLife: Acommunity information management platform for the database research community//P. DeRose, W. Shen, F. Chen, Y. Lee, D. Burdick, A. Doan, and R. Ramakrishnan. - CIDR Forum, 2007.
7. A Platform for Personal Information Management and Integration//X. Dong and A. Halevy. – CIDR Forum, 2005.
8. Мальцев А.И. Алгебраические системы. – М., 1970. – 392 стр.
9. Шаховська Н.Б. Простори даних: поняття та призначення //Матеріали конференції CSIT-2007. – Львів – 2007. – С.269-277.
10. Шаховська Н.Б. Особливості моделювання просторів даних //Комп'ютерна інженерія та інформаційні технології. Вісник НУ "Львівська політехніка", № 608, ст. 145-154, 2008
11. Шаховська Н.Б. Простір даних області наукових досліджень// Моделювання та інформаційні технології. - Інститут проблем моделювання в енергетиці ім. Пухова «Моделювання та інформаційні технології». – Київ, № 45, С.132-140.


Надійшла до редакції 02.03.2009р.

**МЕДИКОВСЬКИЙ МИКОЛА ОЛЕКСАНДРОВИЧ - доктор технічних наук, професор кафедра автоматизованх систем управління, директор інституту комп'ютерних наук та інформаційних технологій, Національний університет «Львівська політехніка», м. Львів, Україна.**

**ШАХОВСЬКА НАТАЛЯ БОГДАНІВНА - кандидат технічних наук, доцент, кафедра інформаційних систем та мереж, заступник декана базової освіти інституту комп'ютерних наук та інформаційних технологій, Національний університет «Львівська політехніка», м. Львів, Україна.**